Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **5**, September **2015**

Location and Image-Based Plant Recognition and Recording System

Jen-Chang Liu and Tsung-Min Lin

Department of Computer Science and Information Engineering
National Chi Nan University
No. 1, University Road, Puli, Nantou, 545, Taiwan
jcliu@ncnu.edu.tw; findnew19@gmail.com

Received March, 2015; revised May, 2015

Abstract. *The earth supports a rich diversity of plants. However, people seldom know their names. One method of identifying plant species is by referring to illustrated plant handbooks. However, this method is ineffective for people unfamiliar with plant features. To help people identify plants, this study develops an application that automatically identifies leaves by utilizing content-based image retrieval and location information. Background subtraction, the first step in our method, was performed using Otsu, color slicing, and GrabCut methods. The most favorable method was determined by evaluating the image segmentation rate. Subsequently, the plant features were extracted using one- and two-dimensional Fourier descriptors. Twenty images each of 21 plant species were collected indoors. The GrabCut method and one-dimensional Fourier descriptor demonstrated the optimal performance. Plant species were automatically identified through leaf recognition using mobile device. The mobile application records plant information through crowdsourcing. To test the system, we collected 10 outdoor images each of 50 plant species. Location-based search engine assisted plant recognition. The experimental results were 78% accurate, that is, the correct species was the first-ranked suggestion in 78% of the cases.*

Keywords: Plant recognition; Crowdsourcing; GrabCut; Fourier descriptor

1. Introduction.

The earth supports a rich diversity of plants. Although several species of plants are encountered in peoples' everyday lives, few can identify them, because such identification is time- and effort-intensive for nonexperts. Although illustrated plant handbooks can be used to identify plants, they are cumbersome for outdoor use. Plants can also be identified by querying websites, such as the NGA Plant Finder [1] and Agriculture Knowledge Entry in Taiwan [2]. To use these online services, users must input plant features, such as the colors of the flowers, blooming season, and foliage characteristics, rendering these services ineffective for users unfamiliar with plant features; users may input inaccurate information and receive erroneous results.

In this study, we designed a mobile application that automatically recognizes plant species using the images of leaves input by the users and the user location. To increase recognition accuracy, users are requested to photograph the leaf on a simple background. Subsequently, users roughly outline the leaf contour using their touch screens for background subtraction. The application extracts and searches the one-dimensional (1D) Fourier description of the leaf in a geotagged plant database. The system retrieves location-specific plant data from the database, performs feature matching, and lists the results in the order of feature matches. After the user selects the most similar plant species from the list, the plant's name, images, and geoinformation are uploaded, thus accumulating more geotagged plant information in the database and enhancing the accuracy of the application. The plant recognition and recording constitute a cycling ecosystem that increases the system richness and performance. This system thus implements crowdsourcing [12] and citizen science [13] in ecological field studies. The operational procedure of our system and the implemented subsystems are detailed in the following sections.

2. **Related Works.** This study crowdsourced plant information through an automatic leaf-image-based plant recognition system. The concept of crowdsourcing and its application is discussed in Section 2.1, and related works on leaf recognition are described in Section 2.2.

2.1. **Crowdsourcing.** Because of the variety and ubiquitous distribution of plants, conducting large-scale ecological field studies is difficult. Many unknown plant species exist, and plant conservation becomes difficult because of the lack of comprehensive plant distribution information, which is referred to as *taxonomic impediment* in biodiversity conservation [3]. The proposed plant recognition and recording system is a type of crowdsourcing for ecological field studies. Crowdsourcing was first promoted in Howe's article in 2006 [12]. Tasks requiring considerable human effort can be outsourced to willing participants, who engage because of either their personal interests or the rewards offered. *Amazon Mechanical Turk* is a typical crowdsourcing platform providing customized services [6]. Examples of practical applications of outsourcing or crowdsourcing include *gengo* [5] and *threadless* [6]. *Gengo* is a website providing language translation services; qualified translators are assigned translation jobs and paid ¥5 per word. *Threadless* is an online clothing and accessory shop that incentivizes designers who participate in their design campaigns through awards and products.

These examples clarify that for success, a crowdsourcing application must incentivize or adequately motivate potential participants. The proposed plant recording system realizes this objective by offering users a novel plant recognition system. Users can quickly identify the correct or closest answer from the ranked list returned by the automatic recognition system. By recording and identifying plants, users familiarize themselves with surrounding plants and also contribute to the database.

2.2. **Leaf Recognition.** The image-based leaf recognition is implemented by using different features, such as leaf curvature [15] and morphology [17]. Kumar et al. [15] used histograms to extract features of the leaf's contour over multiple scales. The leaf samples were manually collected and flattened, and images were captured in a controlled laboratory environment to create a database containing 184 plant species and 23,915 images. Their program was tested using 5,192 outdoor leaf images, and the correct match was within the top five results for 96.8% of the queries.

Nikesh et al. [17] used morphological features, such as leaf width factor, diameter, major axis, minor axis, area, perimeter, form factor, rectangularity, narrow factor, perimeter ratio of diameter, and contour. Their database included 10 plant species and 30 images for each species. Combining different features for matching realized an average recognition precision of 85%-90%.

These studies present opposite research directions. Kumar et al. extracted complex features for leaves and obtained a more favorable recognition performance. A single recognition request requires 5.4 s on an Intel Xeon machine with two quadcore processors running at 2.33 GHz and 16-GB RAM, whereas Nikesh et al. used simpler morphological features that were computed on mobile devices at lower computational power. Contrastingly, this study utilized efficient and effective features computable on mobile devices. To compensate for the precision loss, we used a location-based search, which increased both matching speed and precision.

3. **System Architecture.** This study implemented a mobile-device-based plant recognition and recording system. This system utilizes the open-source data from the Taiwan Biodiversity Network (TBN) [7], an organization supported by the Taiwan Endemic Species Research Institute. TBN provides application programming interfaces (named i35 APIs) [8] that access plant distribution over certain geographical ranges, which helps realize location-based plant recognition. Photographs and geographical information recorded by users are uploaded to the TBN database to enrich its collection. Users can identify plants from anywhere. Users' interests in plant ecology will be evoked if they can acquire knowledge ubiquitously.

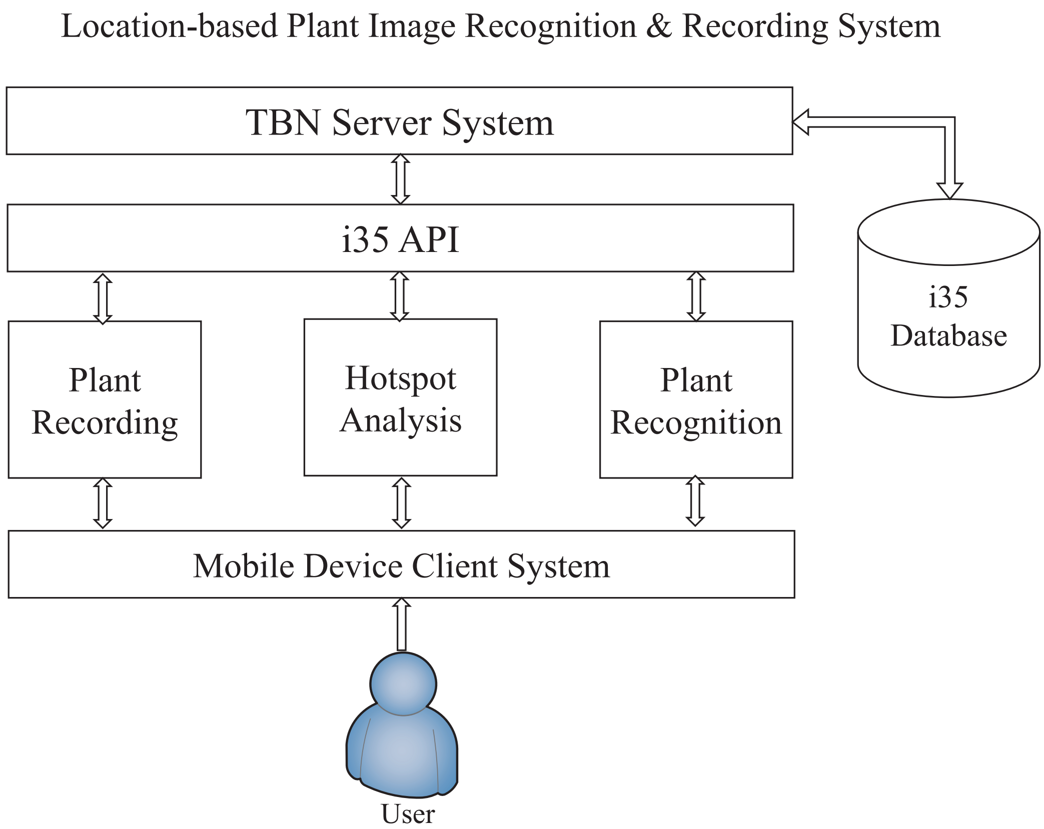


Figure 1. System architecture

The system architecture (Fig. 1) consists of a frontend client and backend server. The client side is implemented on Android OS, and the server side is implemented on the TBN server. Both sides communicate through the i35 APIs, and data are stored in the i35 database. The system consists of the following 3 subsystems:

Plant recognition A user can either capture a photograph of the leaf or select an image from the gallery. Next, the user contours the leaf using the touch screen, and background subtraction is processed using the user-drawn contour. Subsequently, the 1D Fourier descriptor is extracted from the background-removed leaf shape. During image processing, another thread queries the i35 database on the nearby plant species. Feature matching is restricted to target features in the same geographical region as the user, which is detected using GPS data. A list, ranked according to feature matches, is presented to the user.

Plant recording The client interface provides options to capture photographs of different parts of the plants, such as the whole plant, flower, leaf, stem, seed, and root. Moreover, users can add tags explaining the features of each image, such as the shape, margin, and arrangement of leaves. These stored tags can help develop a multimodal plant search in the future. In addition, users can view their plant collections in the form of a photobook.

Hotspot analysis Hotspot analysis provides visualization of the geographical distributions of the user's plant collection. The motivation of a map-based visualization is similar to that of the widely-used application *Fog of world* [9], which visualizes the user's GPS tracks on a world map: users are motivated to add plant records on viewing their collections on a map.

The flowchart of these subsystems is shown in Fig. 2.

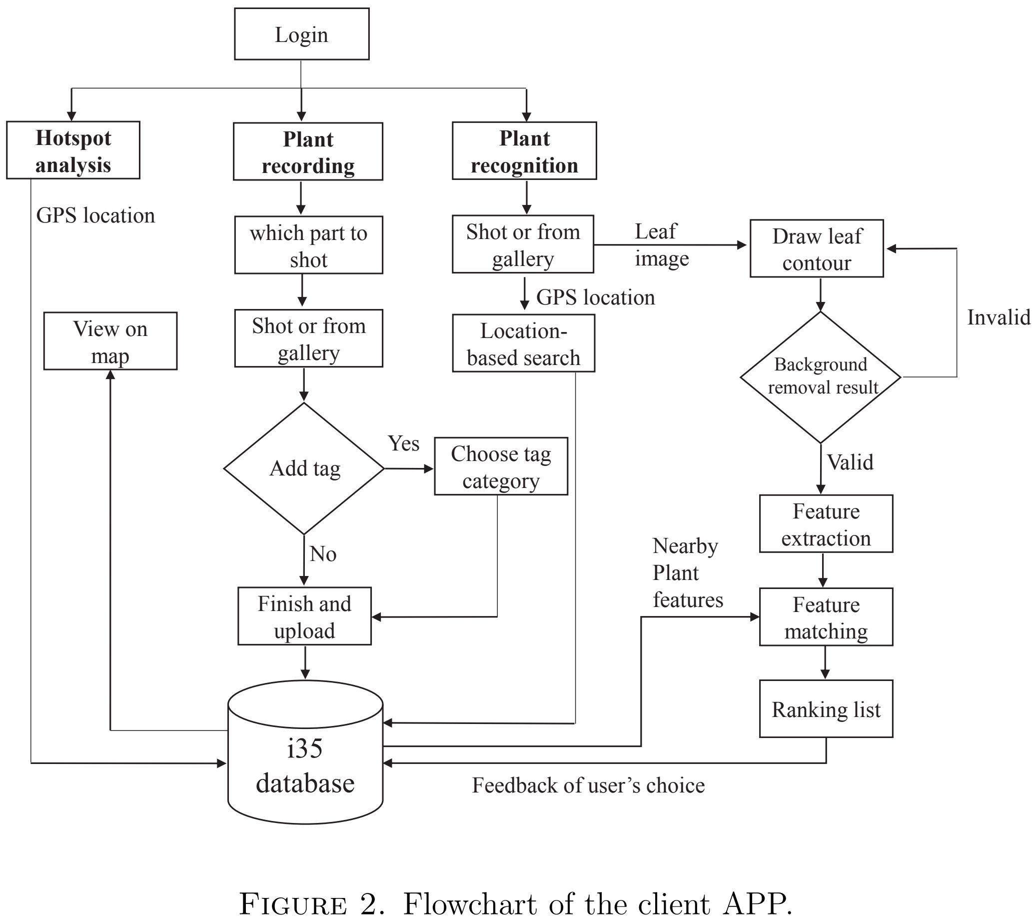


Figure 2. Flowchart of the client APP.

4. **Plant Leaf Recognition.** A leaf photographed in its natural environment includes a cluttered background. Moreover, the direction, position, and viewing angle of the leaf in the image affect feature extraction, which in turn affects matching precision. To remove the background, different background subtraction methods were evaluated in this

study. Feature extraction was performed on the segmented leaf region. Contour-based features were 1D and two-dimensional (2D) Fourier descriptors, which are introduced in Section 4.2.

4.1. **Background Subtraction.** Background subtraction separates the leaf from its background in the target image. Performances of three background subtraction methods are evaluated in this paper: the Otsu method [16] segments gray-level images, the color slicing method [18] processes images directly in the color space, and the GrabCut method [10] combines color and edge features for interactive segmentation. The following sections describe these methods, and the experimental results are presented in Section 5.

4.1.1. *The Otsu method.* The Otsu method was proposed by the Japanese researcher N. Otsu in 1979 [16]. It is a computationally simple but effective method. The optimal threshold is computed on the basis of the statistics of an image's gray-level values. The criterion of this threshold separates all pixels to two classes, and the intraclass variance is minimum. For each leaf image in the database, the Otsu method computes an optimal threshold. Because most leaves are green and have low reflectance, we assumed that pixels lower and higher than the threshold constitute the foreground the background, respectively.

4.1.2. *Color slicing.* The color slicing method segments the color space into the foreground and background. A simple realization uses a spherical region, as shown in Fig. 3. A prototype color index P must be predetermined. For example, P can be the average color of leaves. The color values inside the sphere of center P and diameter T are foreground pixels, and those outside are background pixels.

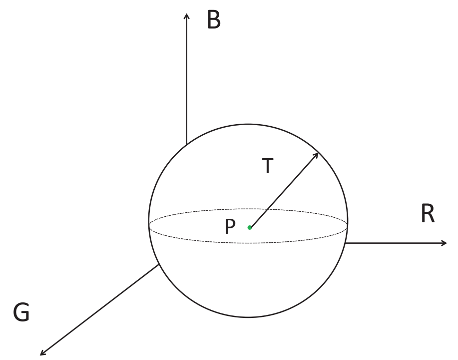


Figure 3. Color slicing in the RGB color space.

4.1.3. *GrabCut.* GrabCut is an interactive background removal algorithm from Microsoft Research [10] currently used in Microsoft Office products. Users draw a mask around the foreground object or choose foreground and background reference points in the target image. To apply GrabCut to our application, two types of masks were considered:

Rectangular mask The user draws a rectangular mask around the leaf region. GrabCut then applies the graph cut algorithm that iteratively approximates the true leaf boundary by using color and edge features. An example is presented in Fig. 4(b). Drawing the mask is easy, but the results may be unsatisfactory.

Custom mask In our implementation, we applied the GrabCut function in the OpenCV library, where a custom mask with an arbitrary shape is defined. Each pixel is assigned four labels: *background, possible background, foreground, and possible foreground.* The assumption is that the user draws the contour slightly outside the leaf. The drawn track is defined as the possible background, the pixels within are defined as the possible foreground, and the pixels outside are defined as the background. An example is presented in Fig. 4(c). This custom mask is applied in GrabCut for background subtraction.

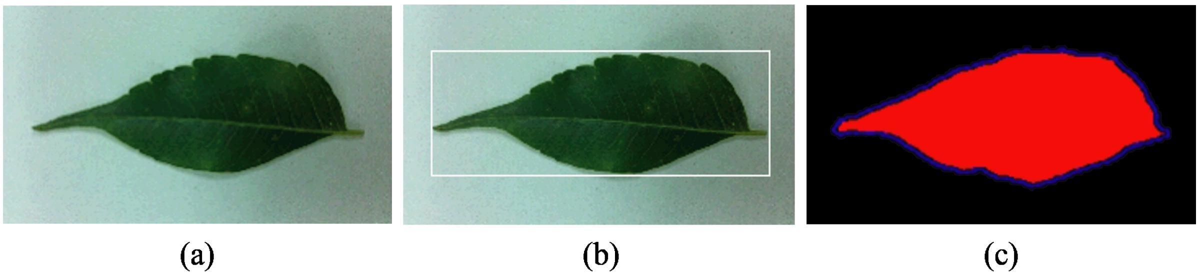


FIGURE 4. Interactive masks using GrabCut. (a) Photograph of a leaf; (b) Rectangular mask; (c) Custom mask.

4.2. **Feature Extraction.** In this study, the plant species were recognized by their leaves' features. A leaf can be distinguished using its features such as apex, base, vein, margins, and petiole. Leaves are typically linear, elliptic, ovate, spatulate, and cordate, and leaf shape is a general feature used for classifying plant species. In this study, we used a 1D Fourier descriptor [11] to represent the 1D trace of the leaf contour. To extract the leaf vein and margin, a 2D Fourier descriptor [19] was applied to represent the 2D features.

4.2.1. *1D Fourier descriptor.* After background subtraction, we obtained the 2D coordinates (x, y) of the leaf contour. These 2D coordinates were transformed to a 1D signal. A 1D Fourier transform was then applied to extract its frequency properties. These Fourier coefficients were used as a feature vector to represent the leaf shape. The detailed processing steps are explained herein.

Step 1: Contour sampling In the collected leaf images, all leaves have different shapes and sizes, and the number of coordinates is not uniform. For comparison, all samples must have the same number of contour coordinates. A higher number yields a more precise leaf shape, but the effects of noise also increase. We evaluated the effect of number of sampling points in our experiments.

Step 2: Centroid distance The sampled 2D coordinates $x(t)$ and $y(t)$, $t = 0, 1, ..., M-1$, are transformed to a 1D signal by computing their distances to the centroid (x_c, y_c). The formula of centroid distance $r(t)$ is:

$$r(t) = \sqrt{(x(t) - x_c)^2 + (y(t) - y_c)^2} \tag{1}$$

The centroid distance is invariant to translation of the leaf in the image because it is normalized with respect to the leaf's centroid.

Step 3: Fourier transform The centroid distance $r(t)$, $t = 0, 1, ..., M - 1$ is processed with Discrete Fourier Transform as follows:

$$R(w) = \frac{\sum_{t=0}^{M-1} r(t) exp(\frac{-j2\pi wt}{M})}{M} \tag{2}$$

The Fourier transform coefficients $R(w)$ represent the phase and weight at each frequency w. Low frequency coefficients represent the rough contour of the leaf and the high frequency coefficients represent the detailed variation of the leaf margin.

Step 4: 1D Fourier descriptor Fourier transform is applied, only half of the Fourier coefficients $R(w)$ are informative. To achieve scale and rotation invariance of the leaf, the 1D Fourier descriptor $\mathbf{F_1}$ was computed using $R(w)$ as follows:

$$\mathbf{F_1} = (\frac{|R(1)|}{|R(0)|}, \frac{|R(2)|}{|R(0)|}, \frac{|R(3)|}{|R(0)|}, ..., \frac{|R(M/2)|}{|R(0)|}) \tag{3}$$

The absolute values of $R(w)$ were used to achieve shift invariance for $r(t)$ and thus rotation invariance of the leaf. By dividing all elements by $|R(0)|$, $\mathbf{F_1}$ becomes invariant to the average strength of $r(t)$, thus becoming invariant to the leaf scale (or size).

4.2.2. *2D Fourier descriptor.* Leaf veins are vascular tissues in the spongy layer, and the central vein is called the midrib; secondary and lateral veins also exist. Various leaf vein patterns are present, such as pinnate, arcuate, parallel, and palmate veins. These features can aid leaf identification. The 2D Fourier descriptor has been used to recognize complex 2D patterns [19]. Before computing the 2D Fourier descriptor, the leaf image must be preprocessed to extract veins and normalized. The processing steps are described herein.

Step 1: Edge detection The background-removed colored leaf image is converted into its corresponding gray-level image, and Laplacian zero-crossing [18] is used to detect vein edge. An example is presented in Fig. 5(b).

Step 2: Position normalization The centers of the leaf and the leaf image are not always aligned. Thus, the leaf centroid is first obtained. According to the motion vectors computed from the leaf centroid to the image center, the leaf edges are moved to the center of the image. This step makes the feature translation-invariant.

Step 3: Polarization To achieve scale and rotation invariance, the original 2D coordinates (x, y) are transformed to polar coordinates (r, θ). Fig. 5(c) shows the polarization of the image presented in Fig. 5(b). Although, the angle ranges from 0° to 360°, the range of the radius varies with leaf size. Therefore, the maximum value of the radius is normalized to 120. Radius normalization results in scale invariance.

Step 4: Fourier transform The polarized image $p(r, \theta)$ is processed using 2D discrete Fourier transform as follows:

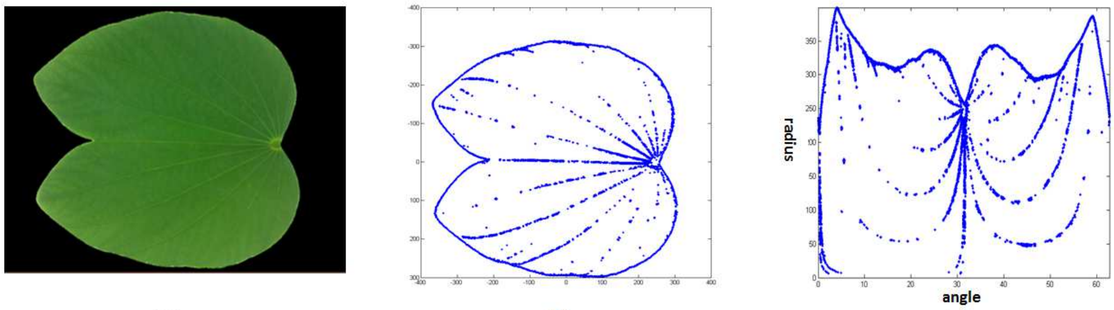
$$P(u, v) = \frac{\sum_{r=0}^{R-1} \sum_{\theta=0}^{T-1} p(r, \theta) exp(-j2\pi(\frac{ur}{R} + \frac{v\theta}{T}))}{RT} \tag{4}$$

The Fourier transform coefficients $P(u, v)$ represent the phase and weight of the 2D frequency patterns that compose the polarized leaf edges.

Step 5: 2D Fourier descriptor High frequency components of $P(u, v)$ are typically small and may contain noise. According to [19], the Fourier descriptor $\mathbf{F_2}$ of length 23 is obtained using the normalized $P(u,v)$ as follows:

$$\mathbf{F_2} = \left(\frac{|P(0,1)|}{|P(0,0)|}, \dots, \frac{|P(0,5)|}{|P(0,0)|}, \frac{|P(1,0)|}{|P(0,0)|}, \dots, \frac{|P(1,5)|}{|P(0,0)|}, \dots, \frac{|P(3,0)|}{|P(0,0)|}, \dots, \frac{|P(3,5)|}{|P(0,0)|}\right) \quad (5)$$

The absolute value of $P(u,v)$ makes the image invariant to θ-directional shift of $p(r,\theta)$, thus achieving rotation invariance.



(a) (b) (c)

Figure 5. 2D Fourier description. (a) Original image; (b) Leaf contour and veins; (c) Polarization of Fig. 5(b).

5. **Experimental Results.** Section 5.1 discusses the plant database used in the experiments. Background subtraction is evaluated in Section 5.2. Finally, plant recognition performance is examined in Section 5.3.

5.1. **Plant database.** As shown in Fig. 2, the GPS coordinates were transmitted to the TBN server to retrieve the distribution of plant species near the user. Location-based search is more effective if complete plant records for the queried region are available. To demonstrate location-based plant search, we surveyed the woody plant distribution in the campus of National Chi Nan University, Taiwan. To avoid dense recording of plant distribution, we recorded one tree for every two neighboring trees of the same species. Photographs, plant species, and GPS location were recorded.

For location-based searching, plant species around the user's position were retrieved. The search region must be appropriately determined. A large region implies more plant species, but more similar plants reduce recognition accuracy. After considering plant distribution density, we set a square region of $0.001° \times 0.001°$ (longitude $\times$ latitude), which equals 102.6 m $\times$ 111.6 m, as shown in Fig. 6.

For the experiments, we collected a dataset, OutDB, containing 50 plant species and 10 outdoor images of each species. Another dataset, InDB, containing 21 plant species and 20 images of each species. The images were captured using smartphones to match the real application-use scenario.

5.2. **Background Subtraction.** An evaluation metric must be defined to assess background subtraction performance. Fig. 7 presents two cases of leaf segmentation. Binary image $\mathbf{A}$ was obtained using a professional photographic software and is termed the "answer," and the result of the automatic background subtraction using the Otsu method, color slicing, or GrabCut is denoted as $\mathbf{B}$. *Precision* and *recall* are defined using the following formulas.

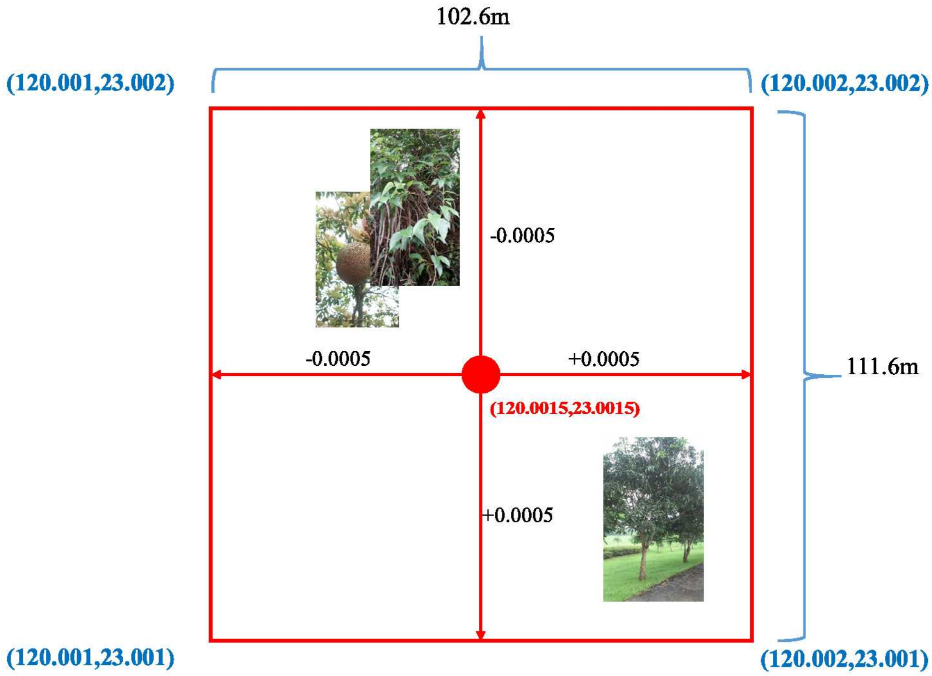


Figure 6. Location-based search.

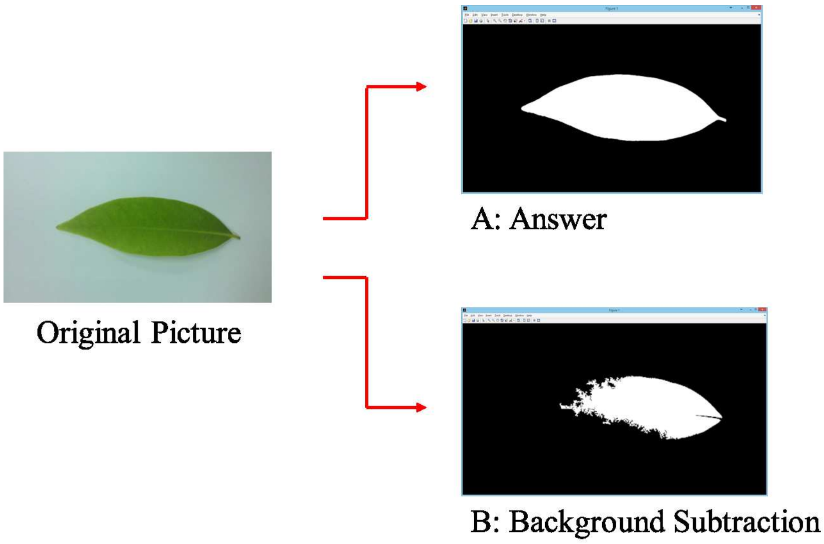


Figure 7. Manual segmentation results (**A**) and background subtraction result (**B**).

$$Precision = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{B}|}, \quad Recall = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A}|} \tag{6}$$

High precision implies that the computed foreground covers more pixels in the answer. As the computed leaf region covers numerous pixels not in the true answer, recall reduces. In data mining, a single measure of accuracy, the *F-measure*, is commonly used.

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

Dataset InDB was used for the background subtraction experiments. F-measures of different background subtraction methods are depicted in Fig. 8. Because the Otsu method processes images only at the gray level, it performs poorly. Color slicing considers color information and thus outperforms the Otsu method. GrabCut outperforms the other two methods because of its complex computing which considers color and edge features. Custom masked GrabCut outperforms the rectangular masked GrabCut. The custom mask is drawn by the user and provides more *a priori* information to the GrabCut algorithm.

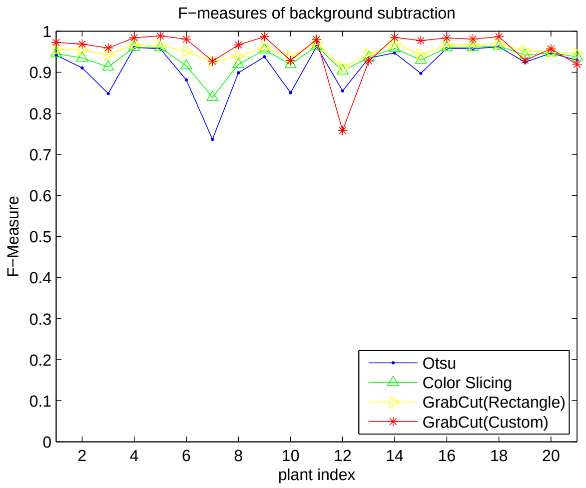


FIGURE 8. F-measures of different background subtraction methods.

5.3. **Plant Recognition.** To avoid the results of background subtraction affecting the evaluation of feature extraction, we use the InDB with manually background subtraction leaf images as the dataset. The query feature is compare with the database features using Euclidean distance. Results are sorting with descending distances. Experiments were carried out by using 5-fold cross validation to separate the InDB as the query set and the database set in each iteration. The performance measure uses *Top N precision*. For the first N plant species in the ranking list, if one of N returns contains the true answer, this query is counted as successful trial.

As mentioned in Section 4.2, the leaf contour must be sampled to a fixed number M in process of 1D Fourier descriptor. After Fourier transform, length of $M/2$ Fourier descriptor is taken as the feature vector. Top N precisions with different sampling points are shown in Table 1.

TABLE 1. 1D Fourier descriptor performance.

Sampling Points	Top 1 Precision	Top 5 Precision	Top 10 Precision
100	60.71%	84.76%	92.38%
150	61.42%	84.04%	92.14%
200	61.19%	83.80%	92.61%

From Table 1, the number of sampling points does not affect the Top N precision. Experiments of more than 200 sampling points have been tried, the results were similar and not shown here. We conjecture that more samples will contain more noise, thus the performance gain by raising the number of samples is reduced by the effects of noise. The highest Top 10 precision is about 92%, it means the user has high chance to find the true answer from the first 10 search returns.

To make a fair comparison with 1D Fourier descriptor, the number of edge points were also sampled to evaluate the performance of 2D Fourier descriptor. Results without sampling were also evaluated. The Top N precisions are shown in Table 2.

Table 2. 2D Fourier descriptor performance.

Sampling Points	Top 1 Precision	Top 5 Precision	Top 10 Precision
100	26.90%	57.14%	76.19%
150	26.19%	64.76%	78.81%
200	31.67%	67.62%	78.33%
no sampling	52%	79%	88%

Since 2D Fourier descriptor takes the leaf contour and vein edges in an 2D image as input, sampling will dramatically reduce the performance. Results without sampling has the best performance. 2D Fourier descriptor performs poorer than 1D Fourier descriptor in all cases. After analysis of the process, we find that performance of edge detection can be the main reason. Detection of edge points on leaf veins can be affected by the lighting condition, thus different number of vein edges points will be detected even for the same plant species.

According to the experimental results of background subtraction and feature extraction, we apply GrabCut with custom mask and 1D Fourier descriptor with 100 sampling points to implement the mobile client system. To simulate the use cases in outdoor environment, the OutDB was used in the experiments. Conditions with and without the location information were examined. Experimental results are shown in Table 3.

Table 3. Location-based plant recognition.

Location-based recognition	Top 1 Precision	Top 5 Precision	Top 10 Precision
Yes	78%	94%	96%
No	28%	58%	64%

The Top 10 precision drops to 64% in outdoor environment. The images taken outdoors have different lighting condition and complex background that make recognition harder. With the help of location information, the precisions raise more than 30% in all cases. The Top 10 precision with location-based search is up to 96%. Query examples with and without location-based search are shown in Fig. 9 and 10. For the same query plant small-leaved mulberry, location-based search returned only two plant species, 5 out of 10 were correct and the first answer was correct. While query without using location-based search, 6 plant species were returned. The correct answer fell to the 4th place.

6. **Conclusions.** This paper proposes to automatically identify the leaves by applying content-based image retrieval assisted with location information. The goal is to investigate appropriate methods that can be implemented on mobile devices with limited computational power and memory. For background subtraction, GrabCut with custom mask has the best performance. A User spends few seconds drawing the contour of the leaf on the touch screen to dramatically increase precision of background removal. 1D Fourier

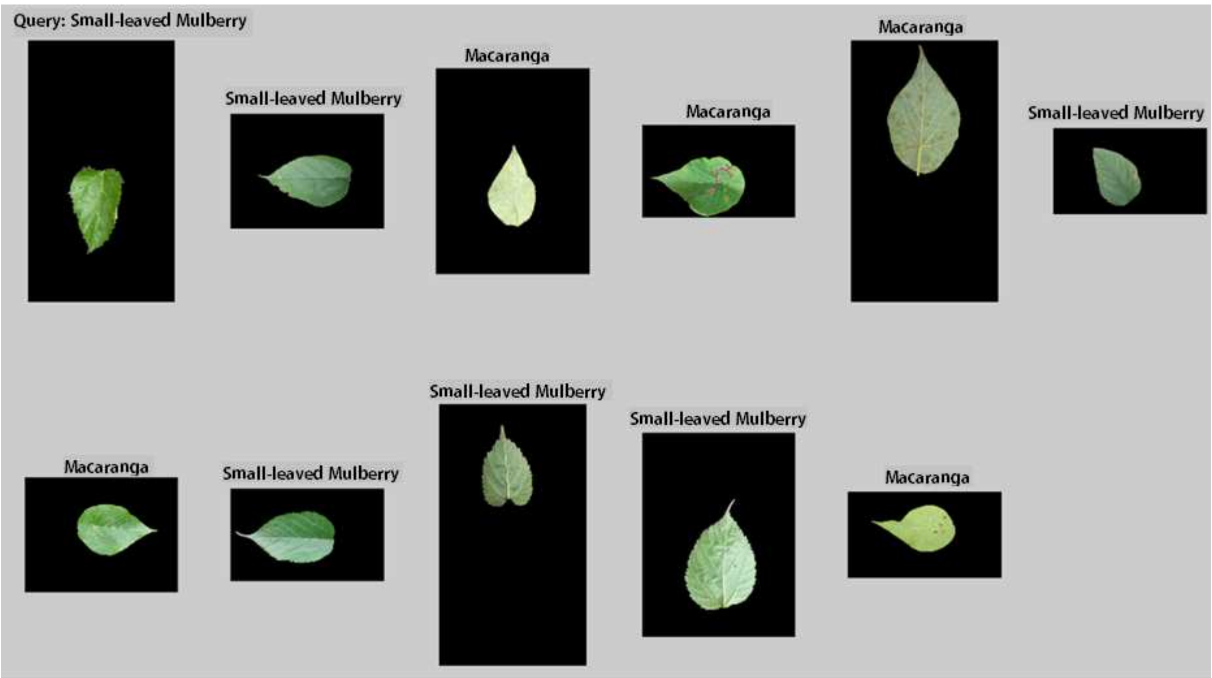


Figure 9. Example with location-based search.

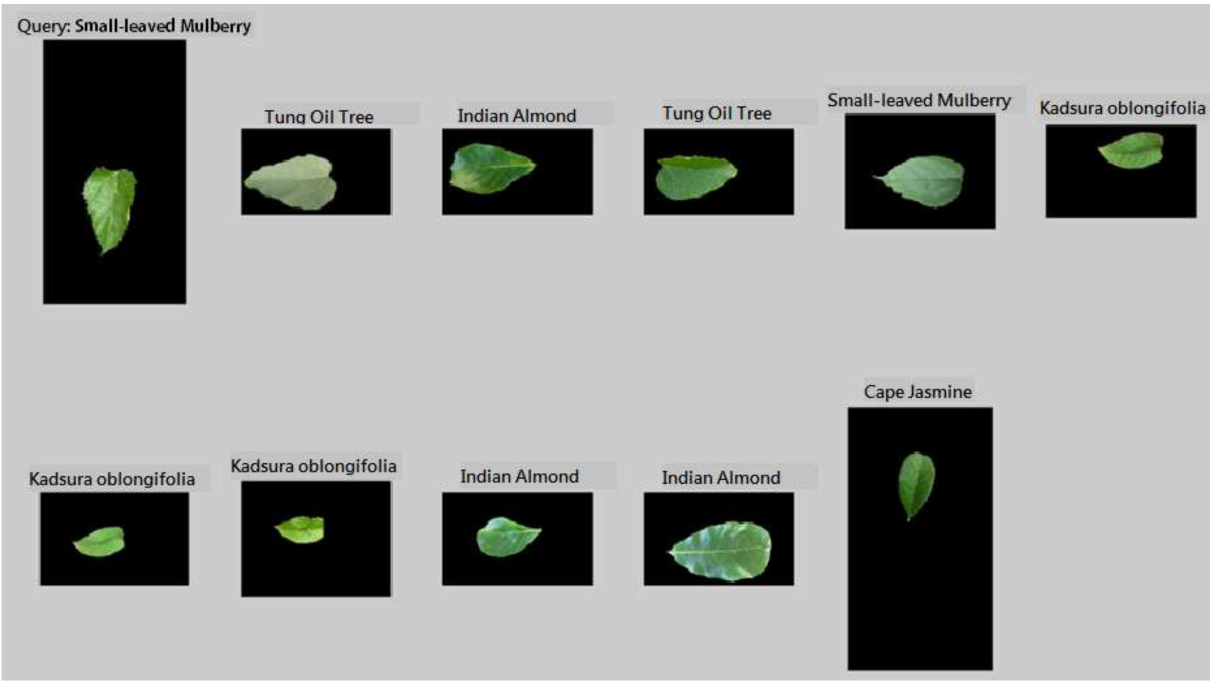


Figure 10. Example without location-based search.

descriptor is efficient and effective to capture the leaf contour feature. The realized mobile APP has been tested outdoors in the campus of National Chi Nan University, Taiwan. By using location-based search, the correct answer appears in the Top 1 position of the ranking list about 78%. The Top 10 precision raises to 96%, which means users can easily find the correct answer by looking over the first 10 returns. The idea of citizen science for ecological field studies is realized by providing people a mobile APP that can reord plant information and help users to identify plant species.

The future work will extend the number of species in our database and add mechanism of relevance feedback to further increase the performance.

Acknowledgment. This work is partially supported by the project MOST 103-2221-E-260-023, Taiwan.

REFERENCES

[1] http://www.garden.org/plantfinder/
[2] http://kmweb.coa.gov.tw/Illustrations/Search.aspx
[3] http://www.cbd.int/gti/
[4] http://www.mturk.com/
[5] http://gengo.com/
[6] http://www.threadless.com/
[7] http://www.tbn.org.tw/Default.aspx
[8] http://www.i35.club.tw/?q=api
[9] http://en.fogofworld.com/
[10] C. Rother, V. Kolmogorov, and A. Blake, Grabcut: Interactive foreground extraction using iterated graph cuts, *ACM Transactions on Graphics*, vol. 23, no. 3, pp.309-314, 2004.
[11] D. Zhang and G. Lu, A comparative study on shape retrieval using fourier descriptors with different shape signatures, in *Proc. of International Conference on Intelligent Multimedia and Distance Education*, pp.1–9, 2001.
[12] J. Howe, The Rise of Crowdsourcing, *Wired Magazine*, Issue 14.06, 2006.
[13] J. L. Dickinson, B. Zuckerberg, and D. N. Bonter, Citizen science as an ecological research tool: challenges and benefits, *Annual Review of Ecology, Evolution, and Systematics*, vol. 41, pp.149–172, 2010.
[14] K. J. Gaston and M. A. O'Neil, Automated species identification: why not?, *Phil. Trans. R. Soc. London*, pp.655-667, 2004.
[15] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares, Leafsnap: A computer vision system for automatic plant species identification, in *Proc. of European Conference on Computer Vision*, LNCS 7573, pp.502–516, 2012.
[16] N. Otus, A threshold selection method from gray-level histograms, *IEEE Trans. Sys., Man., Cyber.*, vol. SMC-9, no. 1, pp.62–66, 1979.
[17] P. Nikesh, P. Nidheesh, and M. S. Shashidhar, Leaf identification using geometric and biometric features, *ASMś International E-Journal of Ongoing Research in Management and IT*, 2013.
[18] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd Edition, 2012.
[19] S. Li, M. C Lee, and D. Adjeroh, Effective invariant features for shape-based image retrieval, *Journal of the American Society for Information Science and Technology*, vol. 56, pp.729–740, 2005.